Weak Post-Compromise Securityを持つIDベース鍵交換技術に関する考察

松井　政裕[1,a)]　飯島　悠介[1]　安田　幹[1]

概要：鍵交換プロトコルにおいて，TPM 等の耐タンパ装置にユーザの秘密鍵を格納した場合，耐タンパ装置の利用権限が不正に利用されると，攻撃者は耐タンパ装置内の秘密鍵を利用して自由に演算を行うことができてしまう．Gordon らは，この耐タンパ装置の利用権限を不正に利用された後の安全性として，Weak Post-Compromise Security(wPCS) を提案している．wPCS では，攻撃者が過去に耐タンパ装置を不正に利用して演算結果を得たとしても，攻撃者が一度利用権限を失えば，その後の鍵交換セッションが安全になる．一方，ID ベース鍵交換技術は ID だけで鍵交換を行うことができるため，機器の数が多い IoT システムや ID を用いたシステムの鍵交換技術として注目されている．本研究では，ID ベース鍵交換プロトコルに wPCS を適応することを目的とし，ID ベース鍵交換プロトコルにおける wPCS に関する安全性モデル (id-wPCS) を提案する．最後に，ID ベース鍵交換プロトコル例として，Hai らの ID ベース鍵交換プロトコルを用いて，どの演算部分を耐タンパ装置で行うと id-wPCS を満たすかを示した．

キーワード：Post-Compromise Security，秘密鍵，漏洩，ID ベース鍵交換，耐タンパ装置

A Study for Weak Post-Compromise Security on ID Based Key Exchange

Masahiro Matsui[1,a)]　Yusuke Iijima[1]　Kan Yasuda[1]

Abstract: When a user using a tamper resistance device such as SIM to store his secret key securely, an adversary can do any calculation using the secret key if the adversary can get access right to use the tamper resistance device. Gordon et al propose Weak Post-Compromise Security (wPCS) which is the security model after adversary gets an access right of a tamper resistance device. On the key exchange protocol that has wPCS, the current key change is still secure if an adversary once had an access right to use the secret key in the tamper resistance device in the past and lost it. In this paper, we estimate the significance of wPCS in the use case of ID Based Key Exchange. Then, we propose Weak Post-Compromise Security for ID Based Key Exchange (id-wPCS). We also propose the example of ID Based Key Exchange Protocol that has id-wPCS.

Keywords: Post-Compromise Security, Secret Key, Leakage, ID Based Key Exchange, Tamper Resistance Device

1. はじめに

鍵交換プロトコルにおいて，一般的にユーザは長期的に使用するユーザごとの秘密鍵（長期秘密鍵）を持っており，これを用いてお互いに正しい鍵交換相手であると認証し合った後、暗号通信等で利用する鍵（セッション鍵）の交換を行う．

したがって，長期秘密鍵が攻撃者に漏洩し，不正に利用されると，攻撃者はユーザになりすますことが可能になり，結果的に攻撃者にセッション鍵を取得されてしまう．そして，セッション鍵を用いて攻撃者は通信内容を盗聴したり，改ざんしたりすることが可能になる．このように鍵交換プ

[1] NTT セキュリティプラットフォーム研究所
NTT Secure Platform Laboratories
[a)] masahiro.matsui.gy@hco.ntt.co.jp

ロトコルにおいて，長期秘密鍵の管理は重要である．

そこで，一般的には TPM(Tamper Persistance Module) などの耐タンパ—性を持つセキュリティ機能をハードウェア (一部ソフトウェア) で実現し，長期秘密鍵を漏洩させないようにする対策が行われている．しかし，TPM などを用いても，攻撃者による長期秘密鍵の不正利用を完全に防ぐ事は難しい．例えば，TPM を使用するにしても，TPM に格納されている長期秘密鍵を利用する権限が攻撃者に与えられれば，攻撃者は長期秘密鍵を使用して，ユーザになりすます事ができる．このように，TPM を利用する際は，TPM の利用権限の管理が重要になってくる．

従来技術では，TPM の利用権限の適切な管理のために，パスワードや公開鍵暗号などを用いた認証技術を利用し，正しいユーザしか TPM を利用できない様にしてきた．しかし，不適切なパスワードの設定や公開鍵暗号の秘密鍵を安全ではない領域で管理するなど，運用面の不備が発生した場合はやはり，TPM の利用権限が攻撃者に渡る恐れがある．

従来の鍵交換プロトコルの安全性に関する研究では，こうした利用権限が攻撃者の手に渡ってしまえば，それ以降の安全性を検討する事はあまりされてこなかった．しかし，鍵交換プロトコルである Signal[2] が，秘密鍵が漏洩した後でも一定の条件下で漏洩後の秘密鍵が安全になるという安全性を持ちはじめ，近年こうした長期秘密鍵が漏洩した後の安全性 (Post-Compromise Security，以下 PCS) の研究が盛んになってきている．また，MLS[3] 等，PCS を持つ鍵交換プロトコルの国際標準化の動きもある．このように，PCS はこれから適応範囲が拡大して行くと考えられる．

一方でこれまでの PCS の研究は，証明書ベースを利用した鍵交換プロトコルに限定されており，利用できる範囲は主に証明書を利用できる範囲に限られてきた．しかし，今後，PCS の適応範囲の拡大を考えた際，例えば IoT 等の分野では証明書の利用が難しいケースがあり，PCS が利用できない可能性がある．岩井らの研究 [7] では，LPWA と呼ばれるナローバンドの通信の利用を想定した場合，互いの証明書を送信するだけで，かなりの時間が掛かると指摘している．そこで，岩井らは IoT では ID ベース鍵交換技術が適しているとしている．ID ベース鍵交換技術は，ID を公開鍵として利用する鍵交換技術であり，ID のみを送信すればよく，LPWA を使用して通信をする IoT 機器に適しているのではないかと考えられている．そこで，本研究では今後 IoT 機器等での利用拡大を見据え，PCS を持つ ID ベース鍵交換プロトコルに関する考察を行う．

1.1 関連研究

以下に，PCS に関する研究と ID ベース鍵交換技術の安全性に関する関連研究を示す．

PCS

PCS に関する研究は，Gordon らの研究 [1] がある．Gordon らは，PCS という安全性の概念を定義し，その定式化を行った．Gorodon らは，weak Post-Compromise Security(wPCS) と full Post-Compromise Security(fPCS) という二種類の PCS をあげている．wPCS は，HSM 等の TPM を用いた際 PCS であり，攻撃者が TPM に格納された長期秘密鍵を用いて演算を行う事ができた後でも，その後のセッション鍵が安全であるという事を保証している．また，fPCS は TPM は用いず，攻撃者が長期秘密鍵を直接取得できた後でも，その後のセッション鍵が安全であることを保証している．現在，IoT の領域ではトラストゾーンやセキュアマイコンなど IoT 機器向けの TPM の利用が進んでいる．そこで，こうした IoT 機器向けの TPM を利用した場合を想定して，本研究では wPCS を対象とする．

ID ベース鍵交換技術

ID ベース鍵交換技術の安全性については，Hai らの id-eCK 安全 [5] がある．これは，認証付き鍵交換プロトコルの一般的な安全性である eCK 安全 [4] を元にして，ID ベース鍵交換技術特有の要素を拡張した安全性である．本安全性では，あるセッション終了後に長期秘密鍵が取得されてもそのセッションのセッション鍵は安全であるを保証している．しかし，セッションの前に長期秘密鍵が取得されてしまうと，その後のセッションのセッション鍵は安全でないとしている．

1.2 wPCS の有用性について

これまで述べてきた wPCS の安全性モデルでは攻撃者の能力に関して強い前提条件を置いている．それは，ユーザ秘密鍵の漏洩後に鍵交換が安全になるためには「攻撃者による TPM 内のユーザ秘密の不正利用後，攻撃者は TPM を利用できない」という条件を満たす必要性がある事である．こうした前提条件がついた安全性は有用であるか検討をする必要がある．

TPM 等を利用した適切な鍵管理については既にガイドライン [8] が存在し，このガイドライン通りに適切に管理を実施をすれば，一般的には不正利用は起こらない．しかし，実際の運用では，不備が起こりうる可能性があり，こうしたケースにおいて，TPM 内のユーザ秘密鍵を不正に利用される恐れがある．この場合，不備を直せば適切な状態になり，攻撃者は TPM 内のユーザ秘密鍵を利用できなくなる．従って，実際のケースでは「攻撃者による TPM 内のユーザ秘密鍵の不正利用後，攻撃者が TPM を利用できない」という条件は不備を修正する事で満たされる可能性が高いと考えらる．すると，一度，不備を修正さえすれば，その後例え攻撃者が既に取得した長期秘密鍵を利用し

た演算結果を使っても，攻撃が出来なくなり，TPM の不正利用をされる前の安全な状態に戻す事ができる．

一方で，wPCS を持たない場合は，例え不備を修正した後でも，既に攻撃者が取得した長期秘密鍵を利用した演算結果を使えば，攻撃が可能である．すると，いくら不備を修正しても安全性は損なわれたままである．このように，実運用を考えた際，wPCS の有用性は損なわれないと考えられる．

1.3 貢献

本研究では，従来の研究で提案された wPCS に対して，ID ベース鍵交換プロトコルの特徴を含んだ新しい安全性 id-wPCS を提案する．その後，ID ベース鍵交換プロトコルの例として Hai らの ID ベース鍵交換プロトコルを紹介し，Hai らの ID ベース鍵交換プロトコルにおいて，どの演算部分を HSM で実行するように設計すると id-wPCS が満たされるかを示す．

2. 記法

本研究では，集合 S から一様ランダムな確率分布に従い，元 $a \in S$ を出力することを $a \leftarrow_R S$ と表記する．次に，以下の仮定を定義する．

定義 1 CDH 仮定

$\mathbb{G}$ を位数 q の巡回群とし，$P \in \mathbb{G}$ を $\mathbb{G}$ の生成元とする．$x, y \in \mathbb{Z}_q$ とし，k をセキュリティパラメータとする．攻撃者 A を Probabilistic Polynomial Time(PPT) Turing Machine とする．また，関数 $\epsilon(k)$ が k に関して無視できるとする．

$$Adv_A^{CDH}(k) := Pr[Z = xyP | A(\mathbb{G}, q, P, X = xP, Y = yP) = Z] \leq \epsilon(k)$$

上記の不等式が成り立つと仮定することを CDH 仮定とする．

定義 2 CBDH 仮定

$\mathbb{G}, \mathbb{G}_t$ を位数 q の巡回群とし，$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$ を対称双線形写像とする．$P \in \mathbb{G}$ を $\mathbb{G}$ の生成元とする．$x, y, w \in \mathbb{Z}_q$ とし，k をセキュリティパラメータとする．攻撃者 A を PPT Turing Machine とする．また，関数 $\epsilon(k)$ が k に関して無視できるとする．

$$Adv_A^{CBDH}(k) := Pr[Z = xywP | A(\mathbb{G}, \mathbb{G}_t, q, P, X = xP, Y = yP, W = wP) = Z] \leq \epsilon(k)$$

上記の不等式が成り立つと仮定することを BDH 仮定とする．

定義 3 DBDH 仮定

$\mathbb{G}, \mathbb{G}_t$ を位数 q の巡回群とし，$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$ を対称双線形写像とする．$P \in \mathbb{G}$ を $\mathbb{G}$ の生成元とする．$x, y, w \in \mathbb{Z}_q$ とし，k をセキュリティパラメータとする．攻撃者 A を PPT Turing Machine とする．また，関数 $\epsilon(k)$ が k に関して無視できるとする．

$$Adv_A^{DBDH}(k) := |Pr[A(\mathbb{G}, X = xP, Y = yP, W = wP, Z = e(P,P)^{xyw})] - Pr[A(\mathbb{G}, X, Y, W, Z \leftarrow_R \mathbb{G}_t)]| \leq \epsilon(k)$$

上記の不等式が成り立つと仮定することを DBDH 仮定とする．

3. id-wPC モデルについて

本章では、Hai らの ID ベース鍵交換のモデル (id-eCK モデル)[5] を参考に，Gordon らの wPCS[1] を含んだモデルを作成する．本モデルを id-wPC モデルとする．

3.1 ユーザと鍵発行センタ

あるシステムの ID の有限集合を $ID = \{id_1, ..., id_n\}$ $(n \in \mathcal{N})$ とする．ユーザの有限集合を $U = \{u_i\}_{i \in ID}$ とし，u_i はそれぞれ長期秘密鍵 (ssk_i) と長期公開鍵 (spk_i) を所有している．U の長期秘密鍵の集合を ssk_U，長期公開鍵の集合を spk_U と表記する．各ユーザは PPT Turing Machine とする．

また，ID や長期秘密鍵の生成を管理する鍵発行センタを KGC とし，KGC はマスターシークレットキー (msk) を所有している．KGC も PPT Turing Machine とする．

3.2 インスタンス

本モデルでは，2 人のユーザ間 u_i, u_j $(i, j \in ID)$ で鍵交換を行う．u_i は u_j と鍵交換を行う際，インスタンス $\pi_{i,j}^{k_i}$ $(k_i \in \mathcal{N})$ を生成する．$\pi_{i,j}^{k_i}$ は，ユーザ i のインスタンスであって，鍵交換相手のユーザ j のインスタンスとの間で k_i 回目の鍵交換を行うことを示している．なお，表記簡略のため，以後 k_i を k と表記する．

鍵交換を行う際，$\pi_{i,j}^{k}$ は役割を持つ．鍵交換プロトコルの最初のメッセージを送付するインスタンスの役割を Initiator とし，I と表記する．Initiator の最初のメッセージを受け取るインスタンスの役割を Receiver とし，R と表記する．

インスタンスにより実施される鍵交換をセッションと呼ぶ．インスタンス $\pi_{i,j}^{k}$ はセッション $s_{i,j}^{k}$ を実行し，以下の情報を所有している．

$actor_{i,j}^{k}$
　インスタンスのユーザ．$actor_{i,j}^{k} \in U$

$role_{i,j}^{k}$
　インスタンスの役割．$role_{i,j}^{k} \in \{I, R\}$

$peer_{i,j}^{k}$
　鍵交換相手のインスタンスのユーザ．$peer_{i,j}^{k} \in U$

$sk_{i,j}^k$
交換されたセッション鍵.

$esk_{i,j}^k$
セッションで利用する短期秘密鍵.

$epk_{i,j}^k$
セッションで利用する短期公開鍵.

$status_{i,j}^k$
セッションのステータス. $status_{i,j}^k \in \{active, accept, reject\}$

$sent_{i,j}^k$
セッションでインスタンスが送ったメッセージ.

$recv_{i,j}^k$
セッションでインスタンスが受け取ったメッセージ.

3.3 セッション

セッションのステータス $status_{i,j}^k$ の各フェーズの定義は以下の通り.

active
$\pi_{i,j}^k$ がユーザにより生成されたら，$\pi_{i,j}^k$ のセッションは *active* になったという.

accept
鍵交換に成功した場合 ($sk_{i,j}^k$ が導出された際)，そのセッションは *accept* したという.

reject
$sk_{i,j}^k$ が導出されず，鍵交換プロトコルが終了した際，セッションは *reject* したという.

なお，$status_{i,j}^k$ が *accept* か *reject* のどちらかになったセッションを *complete session* と呼ぶ.

また，以下の関係を持つセッションをそれぞれ *matching session* と呼ぶ.

定義 4 *matching session*[9]
complete session であるセッション $s_{i,j}^k$，セッション $s_{j,i}^k$ が互いに *matcing session* であるとは，以下の条件が成り立つ事を指す.
$actor_{i,j}^k = peer_{j,i}^k \wedge peer_{i,j}^k = actor_{j,i}^k \wedge sent_{i,j}^k = recv_{j,i}^k \wedge recv_{i,j}^k = sent_{j,i}^k \wedge role_{i,j}^k \neq role_{j,i}^k$

3.4 攻撃者とクエリ

攻撃者は PPT Turing Machine とし，オラクルを利用して以下のクエリを実行する事ができる.

$EstablishParty(id_a)$:
KGC はクエリで指定された $id_a (a \in N)$ を持ったユーザをシステムに登録し，長期秘密鍵 ssk_{id_a} を返す．ただし，攻撃者は既存のユーザを追加する事は出来ないとする．($id_a \notin ID$) なお，このクエリで登録されていない既存のユーザを *honest* なユーザと呼ぶ.

$Create(i, r, j)$:
ユーザ u_i に $role_{i,j}^k = r (r \in \{I, R\})$ として，ユーザ u_j と鍵交換を開始させる.

$Send(\pi_{i,j}^k, m)$:
$\pi_{i,j}^k$ に対してメッセージ $m \in M$(メッセージ空間) を送信し，その返答を受け取る.

$SessionKeyReveal(\pi_{i,j}^k)$:
accept したセッション $\pi_{i,j}^k$ の $sk_{i,j}^k$ を取得する.

$EphemeralKeyReveal(\pi_{i,j}^k)$:
セッション $\pi_{i,j}^k$ の $esk_{i,j}^k$ を取得する.

$StaticKeyReveal(u_i)$:
ユーザ u_i の ssk_i を取得する.

$HSM(u_i, m)$:
ユーザ u_i の ssk_i とメッセージ m を使用して行った演算結果を取得する.

$KGCStaticKeyReveak()$:
KGC の msk を取得する.

$Test(\pi_{i,j}^k)$:
$\pi_{i,j}^k$ をテストインスタンスに指定する.

$Guess(b')$:
$b' \in \{0, 1\}$ をチャレンジャーに渡す.

4. id-wPCS について

前述の id-wPC モデルを用いて，ID ベース鍵交換プロトコル $\mathbb{P}$ に対する id-wPCS を以下の様に定義する.

4.1 id-wPC ゲーム

id-wPC モデルにおいて以下のゲーム ($Game^{id-wPC}$) を定義する.

定義 5 $Game^{id-wPC}$
チャレンジャーは，ユーザ U を生成し，それぞれのユーザの長期秘密鍵 ssk_U と長期公開鍵 spk_U を生成する．攻撃者は U と spk_U を与えられる.
攻撃者はオラクルに対してクエリを行い，その回答を得る．攻撃者はあるタイミングで $Test$ クエリをチャレンジャーに対して送り，あるインスタンス $\pi_{i,j}^k$ を指定する．この時選ばれたインスタンスをテストインスタンスと呼ぶ.
チャレンジャーは $\pi_{i,j}^k$ が $fresh$ であるならば，$b \leftarrow_R \{0, 1\}$ を選ぶ．もし，$b = 0$ であるならば，チャレンジャーはセッション鍵空間 SK_{space} からセッション鍵を一様ランダムに選び，攻撃者に出力する．もし，$b = 1$ であるならば，チャレンジャーは $\pi_{i,j}^k$ のセッション鍵 $sk_{i,j}^k$ を攻撃者に出力する．チャレンジャーによる出力を sk^* と表記する．攻撃者は再びクエリを行い，b を推測する．推測結果 b' を $Guess$ クエリでをチャレンジャーに渡す.

この $Game^{id-wPCS}$ での実験を以下のように表記する.
$Exp_{\mathbb{P},A}^{id-wPCS_0}(k)$:
set U, ssk_U, spk_U

$(\pi^k_{i,j}, s) \leftarrow A_1^{O_1}(U, spk_U)$;
$sk^* \leftarrow_R SK_{space}$;
$b' \leftarrow A_2^{O_2}(sk^*, s)$;
output b';

$Exp^{id-wPCS_1}_{\mathbb{P},A}(k)$:
set U, ssk_U, spk_U
$(\pi^k_{i,j}, s) \leftarrow A_1^{O_1}(U, spk_U)$;
$sk^* := sk^k_{i,j}$;
$b' \leftarrow A_2^{O_2}(sk^*, s)$;
output b';

4.2 Freshness

id-wPC 安全における $freshness(freshness^{id-wPCS})$ を以下の通りに定義する．*honest* なユーザ u_i と u_j によってつくられた *complete session* $s^k_{i,j}$ に対して，以下の条件全てが満たされている際，セッション $s^k_{i,j}$ は $fresh$ であるという．セッション $s^k_{i,j}$ の *matching session* をセッション $s^k_{j,i}$ とする．

(1) $\pi^k_{i,j}$ または $\pi^k_{j,i}$ が $SessionKeyReveal$ を受けていない．
(2) $actor^k_{i,j}$ が $StaticKeyReveal$ を受けておらず，かつ $\pi^k_{i,j}$ が $EphimeralKeyReveal$ を受けていない．
(3) $actor^k_{j,i}$ が $StaticKeyReveal$ を受けておらず，かつ $\pi^k_{j,i}$ が $EphimeralKeyReveal$ を受けていない．
(4) $s^k_{i,j}$ に対する *matching session* が存在していない場合に，$peer^k_{i,j}$ に対して $StaticKeyReveal$ が行われていない
(5) $s^k_{i,j}$ の $status^k_{i,j}$ が active になった後に，$peer^k_{i,j}$ に HSM を行っていない．

なお，$KGCStaticKeyReveal$ を行った際は，全てのユーザに $StaticKeyReveal$ を行ったことと見なす．

4.3 id-wPCS

id-wPCS を以下のように定義する．ID ベース鍵交換プロトコル $\mathbb{P}$ が id-wPCS であるとは，id-wPC モデルにおいて，いかなる PPT Turing Machine の攻撃者 A に対しても $freshness^{id-wPCS}$ を満たしながら $Game^{id-wPCS}$ を行った場合，以下の定義を満たすことをいう．

定義 6 id-wPCS
$Adv^{id-wPCS}_{\mathbb{P},A}(k)$ を攻撃者 A が b を正しく推測する確率とする．$Adv^{id-wPCS}_{\mathbb{P},A}(k) \leq \epsilon(k)$ である時，id-wPCS であるとする．なお，$Adv^{id-wPCS}_{\mathbb{P},A}(k)$ は以下のように定義できる．

$Adv^{id-wPCS}_{\mathbb{P},A}(k) := |Pr[Exp^{id-wPCS-0}_{\mathbb{P},A}(k) \rightarrow 1] - Pr[Exp^{id-wPCS-1}_{\mathbb{P},A}(k) \rightarrow 1]|$

5. id-wPCS を満たす ID ベース鍵交換プロトコル

Hai らの ID ベース鍵交換プロトコル (以下，id-AKE) が id-wPCS が満たす HSM クエリ構成方法を示す．

5.1 id-AKE について

Hai らの ID ベース鍵交換プロトコルは図 1 の通りである．

このプロトコルでは，ユーザの ID を A, B とし，公開パラメータは $(q, \mathbb{G}, \mathbb{G}_t, e, k, P, Z, H_1, H_2, H)$ となる．ユーザ A の長期秘密鍵 (ssk_A) が d_{A1}, d_{A2} となり，ユーザ A の長期公開鍵 (spk_A) が Q_{A1}, Q_{A2} となる．また，ユーザ A の短期秘密鍵は x となり，短期公開鍵は X となる．$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$ を対称双線形写像とし，$\mathbb{G}$, $\mathbb{G}_t$ を位数 q の巡回群とする．プロトコルの詳細は [5] を参考のこと．

まず，id-AKE における HSM クエリを定義する．id-wPCS において，この HSM クエリをどの様に定義するかが重要である．例えば，HSM クエリを $HSM(u_A, xZ) = xZ + d_{A1}$ と定義すると，攻撃者は $xZ + d_{A1}$ から xZ を取り除き，StaticKeyReveal を行わずに長期秘密鍵を取得できてしまう．今回は，HSM クエリで行われる演算をペアリング演算 e とし，以下の様に定義する．

$HSM_1(u_A, Y, Q_{B1}, xZ)$
攻撃者はユーザ u_A を指定し，演算結果 $e(Y+Q_{B1}, xZ+d_{A1})$ を得る．

$HSM_2(u_A, Y, Q_{B2}, xZ)$
攻撃者はユーザ u_A を指定し，演算結果 $e(Y+Q_{B2}, xZ+d_{A2})$ を得る．

5.2 安全性証明

定理 1 $HSM_1(u_A, Y, Q_{B1}, xZ) = e(Y + Q_{B1}, xZ + d_{A1})$, $HSM_2(u_A, Y, Q_{B2}, xZ) = e(Y + Q_{B2}, xZ + d_{A2})$ とし，$(\mathbb{G}, \mathbb{G}_t, e, P)$ に対して CBDH 仮定と DBDH 仮定が成り立ち，$(\mathbb{G}, P)$ に対して CDH 仮定が成り立ち，H_1, H_2, H をランダムオラクルとすると，Hai らの ID ベース鍵交換プロトコルは id-wPCS を満たす．

証明 1 本章では，id-wPC ゲームを以下の様に変化する．攻撃者が選んだセッションをテストセッション $s^k_{A,B}$ とする．チャレンジャーはゲームの最初に攻撃者が選ぶ，テストセッション $s^k_{A,B}$ を推測する事とする．以下の通り，ゲーム変換を行う．なお，ゲーム i の $Adv^{id-wPCS}_{id-AKE,A}(k)$ を $Pr(G_i)$ と表記する．

Game0
オリジナルゲームとする．

Game1
Game0 から，ゲーム中に行われたセッションのうち，セッションの短期秘密鍵 (x, y) が同じになった場合に

A		B
$x \leftarrow_R \mathbb{Z}_q$		$y \leftarrow_R \mathbb{Z}_q$
	$\xrightarrow{X = xP}$	
	$\xleftarrow{Y = yP}$	
$sid = (X, Y, A, B)$		$sid = (X, Y, A, B)$
$Z_1 = e(Y + Q_{B_1}, xZ + d_{A_1})$		$Z_1 = e(X + Q_{A_1}, yZ + d_{B_1})$
$Z_2 = e(Y + Q_{B_2}, xZ + d_{A_2})$		$Z_2 = e(X + Q_{A_2}, yZ + d_{B_2})$
$Z_3 = xY$		$Z_3 = yX$
$SK = H(Z_1, Z_2, Z_3, sid)$		$SK = H(Z_1, Z_2, Z_3, sid)$

図 1 Hai らの ID ベース鍵交換プロトコル [5]

ゲームを停止する．ゲームのセッション数を n とし，短期秘密鍵のビット長を l *bit* とする．すると，セッション間で同じ短期秘密鍵になる確率は高々 $\frac{n^2}{2^{2l+1}}$ となる．短期秘密鍵のビット長を十分長くすれば，この確率は無視できるほど小さくなる．従って，$|Pr(G_1) - Pr(G_0)|$ は無視できるほど小さい．

Game2

Game1 から，チャレンジャーが攻撃者の選ぶテストセッション $s^k_{A,B}$ とそのマッチングセッション $s^k_{B,A}$ を正しく推測する事が出来ない場合，ゲームを停止する．セッション数の n とすると正しく攻撃者がセッションを推測する確率は $\frac{1}{n(n-1)}$ となる．従って，$Pr(G_1) = \frac{1}{n(n-1)} Pr(G_2)$ となる．

Game3

Game2 から攻撃者が，HSM クエリを使用して，$HSM_1(u_B, X, Q_{A1}, yZ)$ または，$HSM_2(u_B, X, Q_{A2}, yZ)$ を行った場合ゲームを停止する．攻撃者が，これらのクエリを行うためには，X を推測する必要がある．短期公開鍵 X のビット長を l *bit* とし，攻撃者が行えるクエリの回数を q 回とすると，攻撃者が上記の事象を行える確率は $\frac{q}{2^l}$ となる．これも，短期公開鍵のビット長を十分長くすれば，無視できるほど小さくなる．従って，$|Pr(G_3) - Pr(G_2)|$ は無視できるほど小さい．

Game4

Game3 から，攻撃者が，$e(Y + Q_{B1}, xZ + d_{A1})$ と $e(Y + Q_{B2}, xZ + d_{A2})$，または $e(X + Q_{A1}, yZ + d_{B1})$ と $e(X + Q_{A2}, yZ + d_{B2})$ を偽造出来たらゲームを停止する．この場合の確率は Hai らの安全性証明 [5] により，無視できるほど小さいと示されている．従って，$|Pr(G_4) - Pr(G_3)|$ は無視できるほど小さい．

Game5

Game4 から，テストセッション $s^k_{A,B}$ の Z_1, Z_2 をランダムな値に変更する．もし，マッチングセッション $s^k_{B,A}$ が complete しているならば，セッション $s^k_{B,A}$ の Z_1, Z_2 もランダムな値に変更する．DBDH 仮定が成り立つとすると，$|Pr(G_5) - Pr(G_4)|$ は無視できるほど小さい．Game5 では，Z_1, Z_2 がランダムな値になるため，ターゲットセッションおよびマッチングセッションのセッション鍵もランダムな値になる．従って，攻撃者はランダムなセッション鍵を推測する事になり，$Pr(G_5)$ は無視できるほど小さくなる．

以上により，$|\frac{1}{n(n-1)} Pr(G_5) - Pr(G_0)| \leq \epsilon(k)$ となり，$Pr(G_5)$ は無視できるほど小さいので，$Pr(G_0)$ も無視できるほど小さくなる．Game0 はオリジナルゲームなので，$Adv^{id-wPCS}_{id-AKE,A}(k) \leq \epsilon(k)$ となり，id-AKE は id-wPCS を満たす．

6. まとめ

本研究では，ID ベース鍵交換プロトコルに対して，HSM 等の TPM を用いた場合に，ユーザ秘密鍵が不正に利用されても，その後のセッション鍵の安全が満たされる安全性（wPCS）を付加した新しい安全性 (id-wPCS) を定義した．また，具体的な ID ベース鍵交換プロトコルとして，Hai らの ID ベース鍵交換プロトコルを紹介し，Hai らの ID ベース鍵交換プロトコルにおいて，ペアリング演算を TPM 内で行う事により，id-wPCS を満たす事を示した．

参考文献

[1] Katriel Cohn-Gordon, Cas Cremers, Luke Garratt, “On Post-Compromise Security”, 2016 IEEE 29th Computer Security Foundations Symposium (CSF)

[2] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt and Douglas Stebila, ”A Formal Security Analysis of the Signal Messaging Protocol” 2017 IEEE European Symposium on Security and Privacy (EuroS&P)

[3] IETF, “The Messaging Layer Security (MLS) Protocol”, https://datatracker.ietf.org/doc/draft-ietf-mls-protocol/

[4] B. A. LaMacchia, K. Lauter, and A. Mityagin, “Stronger security of authenticated key exchange” , In W. Susilo, J. K. Liu, and Y. Mu, editors, ProvSec, volume 4784 of Lecture Notes in Computer Science. Springer, 2007.

[5] Hai Huang, Zhenfu Cao, “An ID-based Authenticated Key Exchange Protocol Based on Bilinear Diffie-Hellman Problem”, ASIACCS ’ 09, March 10-12, 2009
[6] Dan Boneh, Matt Franklin, “Identity-Based Encryption from the Weil Pairing”, CRYPTO 2001: Advances in Cryptology - CRYPTO 2001 pp 213-229
[7] 岩井 光輝，川口 武瑠，割木 寿将，佐々木 太良，藤岡 淳，鈴木 幸太郎，永井 彰, “非対称 Pairing を利用した ID ベース認証鍵交換: IoT 機器への適用”, 2019 Symposium on Cryptography and Information Security
[8] Elaine Barker, “Recommendation for Key Management, Part 1: General (Revised)”, National Institute of Standards and Technology
[9] Cas Cremers and Michèle Feltz “Beyond eCK: Perfect Forward Secrecy under Actor Compromise and Ephemeral-Key Reveal” ESORICS 2012